Comanducci *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2023) 2023:24
https://doi.org/10.1186/s13636-023-00288-5

EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH **Open Access**

Variational Autoencoders for chord sequence generation conditioned on Western harmonic music complexity

Luca Comanducci[1*], Davide Gioiosa[1], Massimiliano Zanoni[1], Fabio Antonacci[1] and Augusto Sarti[1]

Abstract

In recent years, the adoption of deep learning techniques has allowed to obtain major breakthroughs in the automatic music generation research field, sparking a renewed interest in generative music. A great deal of work has focused on the possibility of conditioning the generation process in order to be able to create music according to human-understandable parameters. In this paper, we propose a technique for generating chord progressions conditioned on harmonic complexity, as grounded in the Western music theory. More specifically, we consider a pre-existing dataset annotated with the related complexity values and we train two variations of Variational Autoencoders (VAE), namely a Conditional-VAE (CVAE) and a Regressor-based VAE (RVAE), in order to condition the latent space depending on the complexity. Through a listening test, we analyze the effectiveness of the proposed techniques.

Keywords Conditional music generation, Deep learning, VAE, Harmonic complexity

1 Introduction

Automatic music composition has always been a topic of interest in several research and artistic fields, such as music, musicology, music philosophy and computer science. It is worthy to cite the *Mozart's dice games* as one of the first attempts to compose music in an automatic fashion. The first experiments in automatic music composition by means of a computer were obtained in the 1950s with "The Iliac Suite" by Lejaren Hiller through a stochastic rule-based system [1]. Successive attempts were mostly based on the use of Markov models in order to perform melody or harmony generation [2, 3] or create improvisation systems [4].

More recently, the adoption of deep learning techniques has led to major breakthroughs in automatic music composition and generation [5, 6], causing a renewed interest in the subject also from an industry point of view, especially for what concerns the creative and content provider aspects. AI-based music composition systems are often applied in game design or for automatic soundtrack composition. While in older games, the music was linearly modified when changing game levels or state, in recently proposed games, which often have a non-linear structure, the number of different scenarios requiring to change or modify the music, make it unfeasible for the composer to write in a reasonable time the amount of music needed for every possible scenario [7]. In this context, it is often useful to apply some kind of conditional music generation technique that is able to vary the generated compositions depending on some high-level parameter such as emotion and style. In this paper, we explore the possibility of conditioning the deep learning-based music generation process, specifically of chord progressions, depending on the perceived harmonic complexity value, a parameter that has been demonstrated as being important in determining whether a musical piece will be liked or not [8, 9].

*Correspondence:
Luca Comanducci
luca.comanducci@polimi.it
[1] Dipartimento di Elettronica, Infomazione e Bioignegneria (DEIB), Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy

The main deep learning methods for music composition can be broadly divided into models that directly generate the raw audio waveform and those that generate symbolic music representations (e.g., MIDI).

With respect to raw audio generation, the first major breakthrough was the WaveNet model [10], based on an autoregressive architecture. This approach was then extended in [11], where a series of Vector-Quantized Variational Autoencoders (VQ-VAEs) trained at multiple time scales are fed into a WaveNet Decoder. The method allowed better long-term structures, but worse audio quality. In [12], a Wave-to-Midi-to-Wave model was proposed taking advantage of using notes as an intermediate representation in order to generate raw audio waveforms.

Deep learning-based symbolic music generation techniques are based on a plethora of architectures such as Long Short-Term Memory networks (LSTMs) [13, 14], bidirectional LSTMs [15], Transformers [16] or Variational Autoencoders (VAE) [17].

The human music composition process consists of a layered set of stages that involves music theory, emotion understanding, and high-level timbral, auditory, and music perception, which have been extensively studied also from a neuroscience perspective [18]. Very frequently, the composition process starts with the aim of inducing a high-level perceptual idea of emotion. "I would like to compose a sad and rough piece," is an example. The layered structure of deep learning methods resulted to be effective in modeling music at different levels of abstraction [19–21]. For this reason, much attention is focused today on the conditional music generation task, that is, varying the generated music according to some high-level parameter, which can be easily understood by a musician. WaveNet [10] is explored as a conditional architecture by giving as input an additional tag such as an additional instrument. In [22] a model able to generate polyphonic pieces with a chosen tonal tension profile, is proposed. Whereas, in [23], the authors present the Groove2Groove model, which is an encoder-decoder network able to perform one-shot style transfer. Transformer architectures [24] are involved in [25] in order to create a deep learning model able to condition the generation process depending on a specific theme. A great number of methods, such as the one proposed in this paper, aim at learning latent representations of the symbolic music data that disentangle the variation with respect to the conditioning element. One of the first proposed works that use a VAE for latent-space modeling is [26]. There, the authors introduced the MusicVAE architecture. The encoder consists of a simple RNN, while the decoder is *hierarchical*. The latter encourages the model to correctly utilize the latent representation. In [27] the authors learn an effective latent space for symbolic style-aware music generation, by applying the concept of adversarial regularization to a VAE and leveraging the music metadata information as a prior for the latent space. The latent space is conditioned with respect to tonal tension in [28], while in [29] with respect to emotions, in both cases in order to generate monophonic music.

In this manuscript, we explore two techniques for conditioning the latent space of a VAE in order to generate chord progressions depending on their harmonic complexity, as defined in the Western music culture [8]. Several definitions of harmonic complexity have been proposed in the literature [30]. We consider the complexity model already proposed in [31], where the authors designed an architecture of a language model of tonal chord sequences, used to model cognitive expectation, and demonstrated its ability to estimate the perceived harmonic complexity through a listening test.

We will specifically use a Conditional Variational Autoencoder (CVAE) [32] and a VAE-based regression model [33], which are two variations of the standard VAE that are able to condition the latent space depending on a selected parameter, which in our case will be the harmonic complexity.

The rest of the paper is organized as follows. In Section 2, we present the necessary background related to music complexity and the model proposed in [34]. In Section 3, we thoroughly describe the dataset of chord progressions used in this paper generated through the model proposed in [34]. In Section 4, we formalize the problem of chord progression generation conditioned on harmonic complexity, while in Section 5, we present the two techniques based on CVAE and RVAE architectures. In Section 6, we present the listening test results aimed at exploring the capabilities of the proposed techniques. Finally, in Section 7, we draw some conclusions.

2 Background on harmonic complexity

In this section, we present a brief introduction related to the concept of complexity in music. We will focus on the concept of harmonic complexity, rooted in the Western music culture, and we will also introduce the complexity representation proposed in [31] that will be used in this paper.

2.1 Research in music and complexity

The term complexity is a very broad concept generally used to describe what is felt as unpredictable or counter-intuitive, it lacks a single universal definition and corresponds to a different meaning depending on the context [35]. In [36] four different definitions of music complexity are proposed: *hierarchical*, which considers the structure of music on several levels, *dynamic*, which focuses on the temporal evolution of the music piece and *information-based*, which applies concepts drawn out from algorithmic information theory.

A great amount of effort in music complexity research was oriented toward linking the complexity of a musical piece and the pleasantness that the listener derives from it. In [37] the individual preference for a piece of music was related to the so-called *arousal potential*, which corresponds to the activity produced in the brain after the listening. This relationship behaves like an inverted U-shape curve and it has been studied in terms of both individual [38, 39] and general population-level preference [40] and has been analyzed in the context of contemporary western music in [41], finding that the U-shaped behavior, not only determines songs' popularity but varies depending on the genre.

2.2 Harmonic complexity

No universal way exists in order to measure the complexity of a musical piece and the various proposed approaches usually work by considering one of the following dimensions: acoustic, structure, timbre, rhythm, melody and harmony. In this manuscript, we will focus on the concept of harmonic complexity, which can be defined as the interaction and arrangement of chords in a musical piece [42]. Grounding our research in the context of Western music culture, we will consider Tonal Harmony [43], where the function of the chords is determined by the relation of their root note with respect to a reference pitch class, denoted as the tonic. Although studies demonstrated that non-musicians have some kind of natural concept of tonic harmony [44], it is important to stress the fact that this concept applies only to Western music culture, since the concept of harmony is often not present in other music cultures [45].

In the literature, several approaches have been proposed for measuring tonal complexity, both from audio data and symbolic representations. In [8], the harmonic complexity is subdivided into three classes: *harmonic rhythm*, based on the rate of chord changes, *harmonic dissonance*, related to the relationship between the notes in a chord and *harmonic evolution*, concerning the dynamic evolution of harmony. The complexity derived from the harmonic evolution of the musical content is strongly related to the expectations developed by the listener, regarding the evolution of the harmonic profile of the musical piece. This class of complexity can be further subdivided depending on how the expectation is formed in the mind of the listener. The *sensory expectation* is generated through low-level audio properties, such as the evolution of pitch, while the *cognitive expectation* is generated through the application of high-level representations of the musical content, such as tonal harmony. Depending on the type of expectation chosen, the techniques used in order to estimate the harmonic complexity differ.

When considering sensory expectations, the computation of the harmonic complexity is often performed through metrics such as the Pitch Class Profile (PCP) [46], denoted also as chromagram, such as in [39, 47]. Techniques based on cognitive expectation, instead, compute the quantity of surprise perceived by the listener in order to estimate the harmonic complexity. This is often done by either directly applying music theory rules or by applying machine learning models. In [48, 49], the authors estimate the complexity by proposing two rules used to compute the harmonic distance between subsequent chords while in [50] the distance is estimated via a machine learning model. Other machine learning-based approaches are based on multiple viewpoints, systems, Hidden Markov Models, or Dynamic Bayesian systems [42, 51].

2.3 Data-driven harmonic complexity estimation from chord progressions

In order to treat the harmonic complexity of chord progressions, we use the method proposed in [34], where a compound language model is used in order to generate chord progressions. The probability attached to each chord in the progression is then related to its cognitive expectation, and thus with the perceived harmonic complexity, via a listening test.

More specifically, the compound model [34] models chord progressions by computing the prediction probability of each chord x_i, given the sequence of n previous chords, denoted as $p(x_i|x_{i-1},\ldots,x_{i-n})$, which for brevity will be written as $p(x_i|x_{i-n}^{i-1})$ in the following. This is done by computing a distribution over chord sequences

$$p(x) = p(x_0)\prod_{i>0} p(x_i|x_0^{i-1}) \tag{1}$$

by training three different language models, namely Prediction by Partial Matching (PPM) [52], Hidden Markov Model(HMM) [53], and Recurrent Neural Networks (RNN) [54]. Once these three models are separately trained, in [34], they are combined in a compound model, by averaging their output through

$$p(x_i|x_{i-n}^{i-1}) = \sum_{m\in\mathcal{M}} \pi_m p_m(x_i|x_{i-n}^{i-1}), \tag{2}$$

where m is one of the three models contained in the set $\mathcal{M}$, π_m is the weight parameter applied to each model and $\sum_{m\in\mathcal{M}}\pi_m = 1$. This procedure is done in order to ameliorate the disadvantages of each model by combining them. In [34], the π_m values were selected through a grid search, by maximizing the chord prediction accuracy in terms of cross-entropy. The best model was found to be the one that has $\pi_m = 0$ for both the RNN and PPM models, while $\pi_m = 0$ for the HMM. Further Details related to the training procedure and architecture are contained in [34].

3 Dataset of chord progressions

In order to ease the reading of the manuscript, we present here the dataset of chord progressions, presented in [55] that was used in order to generate the latent space that allows sampling chord sequences following their perceived complexity value. We first describe the technique through which the progressions were sampled and then we describe the general characteristics of the dataset.

3.1 Sampling chord sequences according to harmonic complexity

In [34], it was proven that there exists a correlation between the probability of the generated chord sequences and the complexity perceived by the listeners. In order to create the dataset used for training the VAEs and generating the complexity-dependent latent space, it is first necessary to describe how it is possible to sample chord sequences using the compound language model proposed in [34]. Normally, the procedure would be to sample one chord at a time, that is extracting x_i from $p(x|x_0^i)$, however, using such a technique, it would be impossible to control the final probability of the whole sequence. We instead follow what was done in [55] and consider a dataset sampled using a combination of *temperature sampling* and *uniform sampling*.

Using temperature sampling, we modify the probability distribution $p(x|x_0^{i-1})$ before sampling each chord x_i as follows

$$p_\tau(x|x_0^{i-1}) = \frac{p(x|x_0^{i-1})^{1/\tau}}{\sum_{x\in\mathcal{X}} p(x|x_0^{i-1})^{1/\tau}}, \tag{3}$$

where τ is the temperature parameter and $\mathcal{X}$ is the set of possible chords. Different τ values cause different effects. While $\tau = 1$ maintains the original probabilities, $\tau \to \infty$ tends to make the distribution uniform, and finally, values $\tau \to 0$ tend to output the most probable chords.

Uniform sampling, instead, each chord progression is sampled from a uniform distribution over the set of possible chords $\mathcal{X}$. This allows the division of the generated progressions into different bins, according to the probability.

First, chord progressions are generated using temperature sampling, which allows to more easily create sequences having all possible probabilities p, then the uniform sampling allows us to split the sequences in non-overlapping bins according to the log probability $\log p(x)$ of the sequences.

Table 1 A subset of the dataset divided by different bins (1–30) representing harmonic complexity

Complexity Bin	$\log p(x)$	Progression	Type
1	– 6.460180	CGFGC	Major
	– 7.026709	CmFmBbEbCm	Minor
2	– 7.064298	CGCGC	Major
	– 8.208561	CmAbFmBbCm	Minor
5	– 10.431535	CE7AmFC	Major
	– 10.956227	CmGFmGCm	Minor
10	– 16.131764	CBA♯AC	Major
	– 16.763926	CmDmAbDmCm	Minor
20	– 26.125212	CA7A♯7D♯C	Major
	– 27.258201	CmC♯mG7AbCm	Minor

3.2 Dataset composition

The dataset used in this paper and generated in [55] consists of 6311 sequences composed of 5 chords each. The types of possible chords are four: major *maj*, minor *min*, 5 or fifth chord (power chord), 7 or seventh chord. Sequences begin with either *Cmaj* or *Cmin* and were forced to end with the same chord (i.e. the tonic) since this avoids biases due to the fact that the ending of a sequence could influence the perceived complexity [55].

Following the uniform sampling, chord sequences are ordered, according to their log probability, into 30 bins, where each bin corresponds to a log-probability interval. In Table 1, we present a few examples of chord sequences related to corresponding complexity bins and log-probability intervals, while in Fig. 1, we present two histograms representing the number of chord progressions per complexity bin, separated for what concerns the major and minor progressions.

It is interesting to note how the different complexity classes correspond to the type of musical associations identified in musical theory literature. If we analyze the major sequences, we can see that progressions in the first bins (low complexity) contain mainly harmonic transitions between chords I-IV-V, respectively: the tonic, the subdominant and the dominant. These types of progressions are usually denoted as simple in terms of complexity [56].

The complexity can then be treated through continuous values, i.e., the log probabilities of the chord sequences or as discrete ones, corresponding to log-probability intervals, the latter denoted as *bins* in the following. We will use the discrete bin representation in the proposed techniques since it enables us to treat the complexity levels as classes. When training the network models, while we used the same span of 30 classes for both of them, we grouped them differently in order to better exploit the characteristics of the models. The

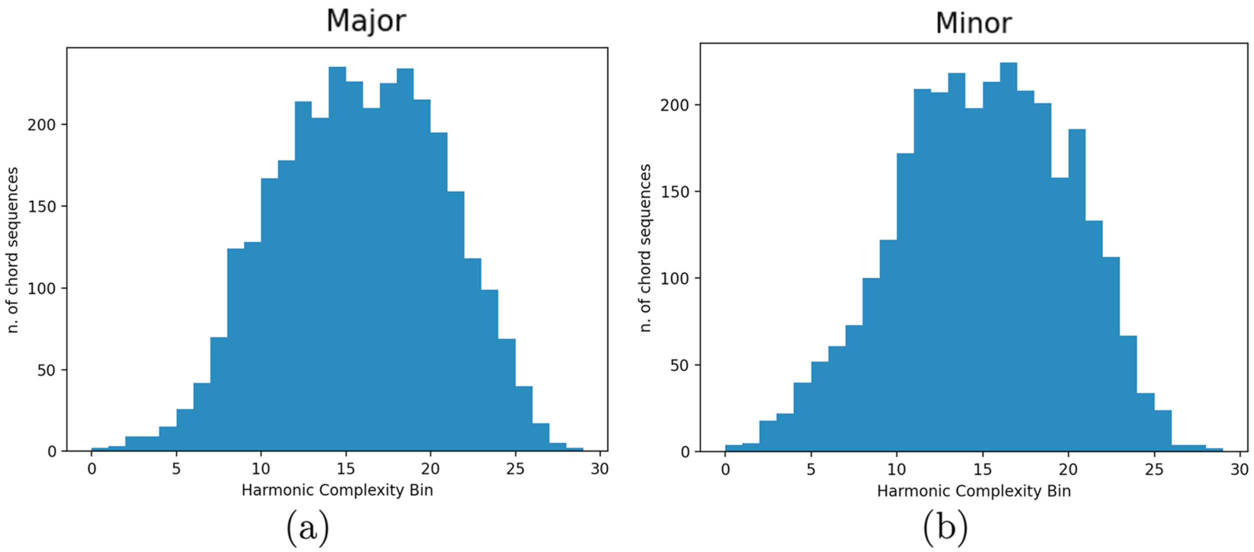
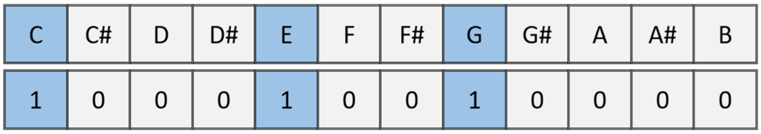


Fig. 1 The number of chord progressions in the dataset per bin of harmonic complexity divided into major and minor sequences. The former (**a**) begins and ends with C maj, while the second (**b**) with C min. The number of harmonic complexity bins is 30



C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1	0	0	0	1	0	0	1	0	0	0	0

Fig. 2 Representation of a C major chord (composed by the notes of C, E, G) in multi-hot format

RVAE was trained using chord sequences with the associated 30 harmonic complexity bins, as in the dataset definition, since this allows the creation of a continuous axis in the latent space capable of modeling this feature from which new progressions were generated. Instead, while training the CVAE, we grouped the 30 complexity classes into 5 classes, each comprising 6 consecutive bins. The 5—classes representation will be denoted as *aggregated bin* in the following. This choice was made because the model generates a latent space associated with each of the complexity classes and not a continuous space as in RVAE. By compacting the classes, the number of data associated with each class is standardized and the latent spaces obtained are able to better capture the properties associated with different values of harmonic complexity.

4 Problem formulation and data representation

In this section, we formalize the goal of the methods proposed in this paper, that is, to learn a latent space that is capable of generating chord sequences using harmonic complexity as a conditioning parameter.

Let us consider sequences of chords represented in the symbolic domain, each chord may be described by a multi-hot vector $\mathbf{x} \in \mathbb{Z}_2^{N_p}$, where $N_p = |\mathcal{P}|$ and $\mathcal{P} = \{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}$ is the set of pitch classes, such that the vector components equal to 1 correspond to the notes composing the chord as shown in Fig. 2. A progression of M chords can then be represented by stacking together M multi-hot $\mathbf{x}$ vectors into a binary matrix $\mathbf{X} \in \mathbb{Z}_2^{M \times N_p}$ as shown in Fig. 3.

Each chord progression can be assigned to a specific value of harmonic complexity [31] through a one-hot vector $\mathbf{c} \in \mathbb{Z}_2^{N_c}$, where N_c corresponds to the number of complexity classes.

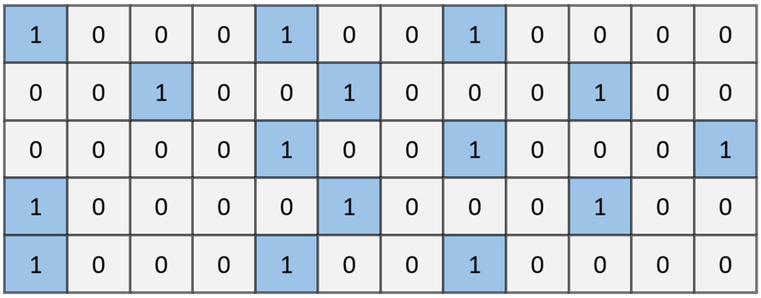
The generation process can then be defined as

$$\mathbf{X} = \mathcal{U}(\mathbf{c}), \tag{4}$$

where $\mathcal{U}(\cdot)$ is a function that is able to generate chord progressions whose complexity is the same as the desired one $\mathbf{c}$ by properly sampling the latent space.

5 Chord sequence generation conditioned on harmonic complexity

In this section, we present two techniques for the generation of sequences of chords conditioned on their harmonic complexity and to model the latent space based on this feature. We first generally present how a Variational Autoencoder (VAE) can be used to generate



1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	0	0	1	0	0	0	0

Fig. 3 The M × N_p matrix representing the chord progression composed by 5 chords. The sequence in this example is C maj, D min, E min, F maj, C maj

chord sequences, and we introduce the notation that will be used throughout the rest of the paper. Then, we present how the generation can be conditioned based on the harmonic complexity through a Conditional Variational Autoencoder (CVAE) [32] and on a VAE-based regression model [33].

5.1 Variational Autoencoder for chord sequence generation

Variational Autoencoders (VAEs) [57, 58] are directed graphical models. Considering the chord progression matrix $\mathbf{X}$ as input to the VAE, a set of latent variables $\mathbf{z}$ is generated from the prior distribution $p_\theta(\mathbf{z})$, while $\mathbf{X}$ is generated by the generative distribution $p_\theta(\mathbf{X}|\mathbf{z})$.

In order to approximate the true posterior $p_\theta(\mathbf{X}|\mathbf{z})$, a proposal distribution $q_\phi(\mathbf{X}|\mathbf{z})$ is introduced. Then the VAE can be formulated through an encoder and decoder network that model the distributions $q_\phi(\mathbf{X}|\mathbf{z})$ and $p_\theta(\mathbf{X}|\mathbf{z})$, respectively, by retrieving the set of parameters ϕ and θ.

Considering a training set of chord sequences $\mathcal{X}^{(train)} = \{\mathbf{X}_1, ..., \mathbf{X}_{N^{(train)}}\}$ the encoder distribution $q_\phi(\mathbf{z}|\mathbf{X})$ is learned such that it is consistent with the posterior $p_\theta(\mathbf{X}|\mathbf{z}) \propto p_\theta(\mathbf{X}|\mathbf{z})p_\theta(\mathbf{z})$ by maximizing the lower-bound of the log marginal distribution of $\mathbf{X}$

$$\begin{aligned} \log p_\theta(\mathbf{X}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{X})}\left[\log p_\theta(\mathbf{X}|\mathbf{z})\right] \\ -D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{X})||p_\theta(\mathbf{z})), \end{aligned} \tag{5}$$

where $D_{\mathrm{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence. Eq. (5) is minimized when $q_\phi(\mathbf{z}) = p_\theta(\mathbf{z}|\mathbf{X})$.

Usually, gaussian latent variables are assumed and $q_\phi(\mathbf{z}|\mathbf{X}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}))$, where the mean $\boldsymbol{\mu}_\phi(\mathbf{x})$ and variance $\boldsymbol{\sigma}^2_\phi(\mathbf{x})$ are obtained as the output of the encoder network.

The second term of Eq. (5) can be marginalized, by assuming Gaussian variables and can be interpreted as a regularization term, forcing the elements of the encoder output to be normally distributed and uncorrelated. The first term, instead is akin to an autoencoder reconstruction error, by applying the so-called reparameterization trick [57] $\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}^2_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ denotes the element-wise product, then $\mathbf{z}$ can be replaced by sampling $\boldsymbol{\epsilon}$, independently of θ, allowing to backpropagate the error through Gaussian latent variables. In this way, the VAE can be efficiently trained through Stochastic Gradient Descent (SGD).

5.2 Conditional Variational Autoencoder

Simply training a VAE using a dataset of chord progressions, would not allow us to control the generation process depending on the harmonic complexity.

Therefore, we propose a technique based on the use of a Conditional Variational Autoencoder (CVAE), that is, a modification of the VAE architecture where a label, in our case the harmonic complexity class, is used to condition the VAE.

Specifically, given the complexity vector $\mathbf{c}$ and the chord progression matrix $\mathbf{X}$, the CVAE consists of an encoder modeling the conditional distribution $q_\phi(\mathbf{z}|\mathbf{X}, \mathbf{c})$ and of a decoder modeling the conditional distribution $p_\theta(\mathbf{X}|\mathbf{z}, \mathbf{c})$ by retrieving the set of corresponding parameters ϕ and θ, respectively. We again assume Gaussian latent variables $q_\phi(\mathbf{z}|\mathbf{X}, \mathbf{c}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}, \mathbf{c}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}, \mathbf{c}))$ and the training can be performed as in the VAE case, where the variational lower bound, defined in (5) becomes

$$\begin{aligned} \log p_\theta(\mathbf{X}|\mathbf{c}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(z|\mathbf{X},\mathbf{c})}\left[\log p_\theta(\mathbf{X}|\mathbf{z}, \mathbf{c})\right] \\ -D_{\mathrm{KL}}(q_\phi(z|\mathbf{X}, \mathbf{c})||p_\theta(\mathbf{z}, \mathbf{c})). \end{aligned} \tag{6}$$

We provide in Fig. 4 a schematic representation of how to use the CVAE model during the training procedure.

5.2.1 Network architecture

The input to the network corresponds to the vectorized chord progression matrix $\mathbf{X}$ concatenated with the complexity vector $\mathbf{c}$, resulting in $[\mathrm{vec}(\mathbf{X})\ \mathbf{c}] \in \mathbb{R}^{MN_pN_c}$, where vec denotes vectorization.

The proposed encoder network is then structured as follows:

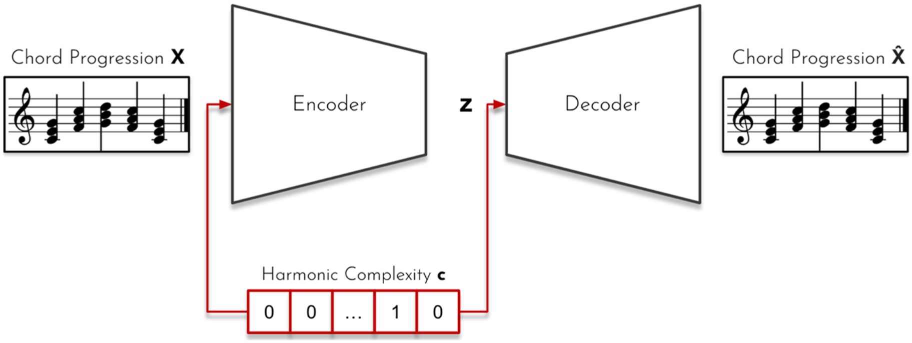

Chord Progression **X**
Encoder
z
Decoder
Chord Progression $\hat{\mathbf{X}}$
Harmonic Complexity **c**
0
0
...
1
0

Fig. 4 Schematic representation of the proposed CVAE architecture

(i) A fully connected layer, with 512 neurons, followed by dropout [59].
(ii) A fully connected layer, with 512 neurons.
(iii) A fully connected layer, with 128 neurons.
(iv) Two parallel fully connected layers, with 2 neurons each generating the mean μ and standard deviation σ.

The latent variable z is then obtained as $z = \mu + \sigma \circ \epsilon$, where ϵ corresponds to random noise and is concatenated with the complexity class vector **c**, before being fed to the decoder, whose architecture is structured as follows

(iv) A fully connected layer, with 128 neurons, followed by dropout [59].
(v) Two fully connected layers, with 512 neurons.
(vi) A fully connected layer, with MN_p neurons.

All fully connected layers are followed by a ReLU activation, with the exception of layers iii and vi where linear and sigmoid activations were used, respectively. We used a dropout rate of 0.2.

5.2.2 Deployment

Once the CVAE is trained, the generation process of chord sequences can be described as follows

- Sample a random latent variable **z** from the prior distribution $p(\mathbf{z})$, i.e., the latent space.
- Concatenate the **z** variable with the **c** conditioning vector of choice and generate a new data from $p_\theta(\mathbf{X}|\mathbf{c}, \mathbf{z})$

Since the output of the network is in the range [0, 1], due to the sigmoid activation, it is necessary to binarize it in order to get values suitable for the chord representation chosen. In order to do this, we applied a simple pattern-matching procedure by computing the cosine distance between each chord in the generated progression and all the 48 possible obtainable chords using the 12 pitch classes and 4 chord types. The chord corresponding to the smallest distance was chosen.

A schematic representation of the chord sequence generation procedure using the CVAE model is shown in the top part of Fig. 5.

5.3 Variational Autoencoder and Regressor

The second method presented in this paper, modifies the VAE architecture by adding a probabilistic Regressor (RVAE) [33] to explicitly condition the data distribution in latent space with respect to harmonic complexity.

The RVAE architecture can be divided into two parts: the *Inference Model* (i.e., the encoder), which estimates the latent representation **z** and the *Generative Model* (i.e., the decoder), whose role is to generate the corresponding chord progression with the desired complexity from the latent vector. The schematic diagram of the model is shown in Fig. 6.

Specifically, given the chord progression matrix **X** and the scalar value $c \in \mathbb{N}^+$ value associated with the harmonic complexity, we assume that the latent representation **z** of **X** is dependent on c. Then, the likelihood distribution underlying each chord progression **X** is $p(\mathbf{X}) = \int_{\mathbf{z},c} p(\mathbf{X}, \mathbf{z}, c)$ and the generative process of **X** can be defined as $p(\mathbf{X}, \mathbf{z}, c) = p(\mathbf{X}|\mathbf{z})p(\mathbf{z}|c)p(c)$, where $p(c)$ is a prior on harmonic complexity. The encoder $p_\theta(\mathbf{X}|\mathbf{z})$ is parameterized as described in Section 5.1. The modeling of the latent representation **z** is different from the standard VAE which uses a single Gaussian prior to generate **z**. In the case of the RVAE, instead, we explicitly condition the latent representation on c so that $p_\theta(\mathbf{z}|c)$ captures an attribute-specific prior on latent representation. We define this part of the model as the *latent generator*, as it can sample a latent representation **z** for a given value of c from this distribution.

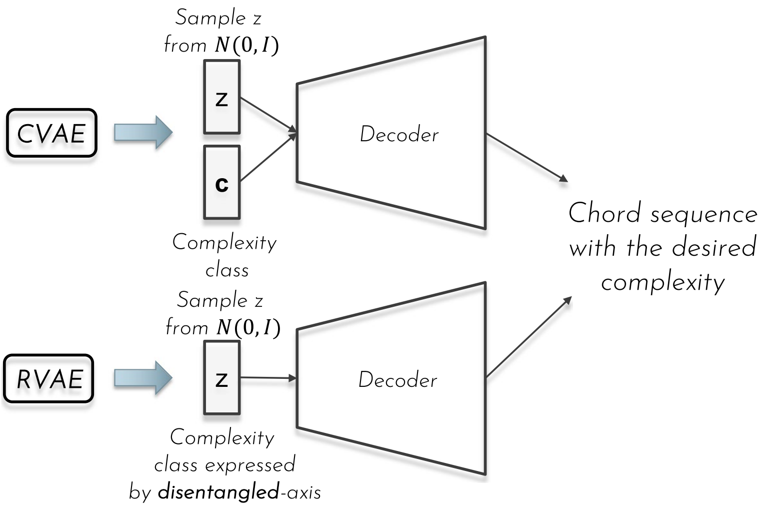


Fig. 5 Schematic representation of the chord generation procedure using the CVAE (top) and RVAE (bottom) models

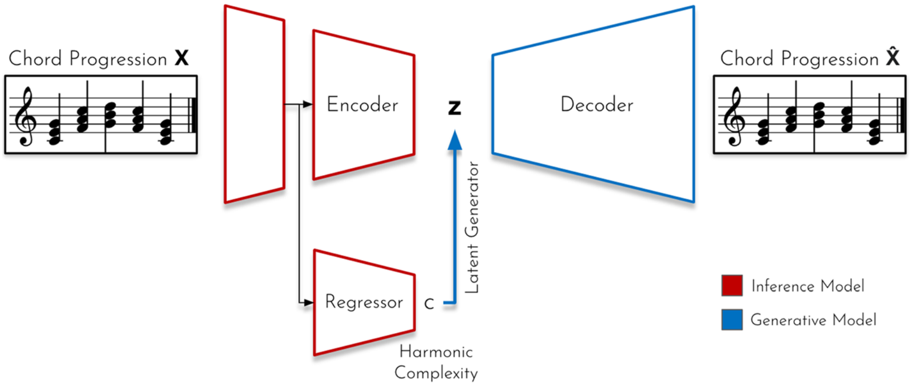


Fig. 6 Schematic representation of the proposed RVAE architecture

As in [33], we assume that the decoder network $p_\theta(\mathbf{X}|\mathbf{z})$ is able to capture the non-linearity of the generative model $p_\theta(\mathbf{z}|c)$, then the latent generator can be parameterized with a linear model

$$p_\theta(\mathbf{z}|c) \sim \mathcal{N}(\mathbf{z}; \mathbf{u}^{\mathrm{T}}c, \sigma^2\mathbf{I}), \mathbf{u}^{\mathrm{T}}\mathbf{u} = 1, \tag{7}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{u}$ is the disentangled dimension [60]; moving along $\mathbf{u}$ yields harmonic complexity-specific latent representations.

The parameters of the RVAE can then be estimated by maximizing the sum of the log-likelihood $\sum_{i=1}^{n} \log p(x_i)$. This maximization is performed through the variational inference procedure and defining an auxiliary function $q_\phi(\mathbf{z}, c|\mathbf{X})$ to approximate the true posterior $p_\theta(\mathbf{z}, c|\mathbf{X})$. We rewrite $\log p(\mathbf{X})$ as:

$$\log p(\mathbf{X}) = D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}, c|\mathbf{X})||p_\theta(\mathbf{z}, c|\mathbf{X})\right) + \mathcal{L}(\mathbf{X}), \tag{8}$$

where $\mathcal{L}(\mathbf{X})$ is the variational lower bound (or ELBO).

Based on mean-field theory, which considers that the behavior of a stochastic model can be approximated by the average value of the elements from which is composed, we assume $q(\mathbf{z}, c|\mathbf{X}) = q(\mathbf{z}|\mathbf{X})q(c|\mathbf{X})$. Then, the ELBO is defined as:

$$\begin{aligned}\mathcal{L}(\mathbf{X}, \phi, \theta) = &-D_{\mathrm{KL}}\left(q_\phi(c|\mathbf{X})||p_\theta(c)\right)+\\ &\mathbb{E}_{(\mathbf{z}|\mathbf{X})}\left[\log p_\theta(\mathbf{X}|\mathbf{z})\right]-\\ &\mathbb{E}_{q_\phi(c|\mathbf{X})}\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{X})||p_\theta(\mathbf{z}|c))\right].\end{aligned} \tag{9}$$

We indicate $q_\phi(c|\mathbf{X})$ as the probabilistic Regressor and formulate it as a univariate Gaussian $q_\phi(c|\mathbf{X}) \sim \mathcal{N}(c; f(x; \phi_c), g(x; \phi_c)^2)$, where ϕ_c are the parameters of the inference networks. The first term in Eq. 9 is the KL divergence, which regularizes the prediction of c with regard to a prior distribution. In our supervised model, the ground-truth of c is known for each training sample $\mathbf{X}$, so this term can be substituted by $\log q_\phi(c|\mathbf{X})$. As for the standard VAE model, the remaining part of the inference involves the construction of a probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{X})$, which maps the input chords progression $\mathbf{X}$ to a posterior multivariate Gaussian distribution in the latent space $q(\mathbf{z}|\mathbf{X}) \sim \mathcal{N}(\mathbf{z}; f(x; \phi_z), g(x; \phi_z)^2 I)$. The second term of Eq. 9 corresponds to the reconstruction loss, which promotes the proper reconstruction of the input data from the latent space, this is similar to what is proposed in the standard VAE architecture. The last condition encourages the posterior $q_\phi(\mathbf{z}|\mathbf{X})$ to resemble the harmonic complexity-specific prior $p(\mathbf{z}|c)$.

All these terms combined concur in linking latent representations with conditional feature prediction. The expectation of the last two terms of Eq. 9 are maximized using the Stochastic Gradient Variational Bayes (SGVB) estimator through the reparameterization trick [57].

We provide in Fig. 6 a schematic representation of how to use the RVAE model during the training procedure.

5.3.1 Network architecture

The input to the network corresponds to the vectorized chord progression matrix $\mathbf{X}$. In the RVAE architecture, the complexity value c is not given as input to the network, since it is directly estimated through the regressor model.

The proposed encoder network is then structured as follows:

(i) A fully connected layer, with 512 neurons, followed by dropout [59].
(ii) A fully connected layer, with 256 neurons, followed by dropout
(iii) A fully connected layer, with 64 neurons
(iv) Two parallel fully connected layers, with 2 neurons each generating the mean μ and standard deviation σ

Both dropout layers have a rate of 0.2. The latent variable $\mathbf{z}$ is then obtained as $\mathbf{z} = \mu + \sigma \circ \epsilon$, where ϵ corresponds to random noise. The Regressor is a regular feed-forward network with an additional output being the uncertainty (i.e. standard deviation) of the prediction.

The architecture of the decoder is structured as follows:

(iv) A fully connected layer, with 64 neurons
(v) A fully connected layer, with 128 neurons
(vi) A fully connected layer, with 512 neurons
(vii) A fully connected layer, with MN_p neurons

All fully connected layers are followed by a ReLU activation, with the exception of layers iii and vii where linear and sigmoid activations were used, respectively.

5.3.2 Deployment

Once the RVAE is trained, the generation process of chord sequences can be described as follows

- Sample a random latent variable $\mathbf{z}$ with the conditioning harmonic complexity bin of choice from the prior distribution $p(\mathbf{z}|c)$, i.e.. the latent space, and generate a new data from $p_\theta(\mathbf{X}|\mathbf{z})$

The same binarization procedure applied using the CVAE can be used with the output of the RVAE. A schematic representation of the chord sequence generation procedure using the CVAE model is shown in the bottom part of Fig. 5.

5.4 Latent space visualization

We show the latent spaces obtained from the input data and the feature labels $q_\phi(\mathbf{z}|\mathbf{X}, \mathbf{c})$ obtained through the CVAE in Fig. 7a, and $q_\phi(\mathbf{z}|\mathbf{X})$, obtained through the RVAE technique, in Fig. 7b. As expected, when using the CVAE, the latent space has no discernible visual behavior, while in the case of the RVAE, the harmonic complexity c is encoded on a disentangled axis in the latent representation of the data. By moving along this axis we are able to generate chord sequences according to their complexity values while being relatively invariant to changes in other factors [33, 61].

6 Experiments

In this section, we present results related to experiments performed in order to demonstrate the capabilities of the proposed techniques in generating chord sequences according to the chosen complexity values. The code used to train the models and generate the chord progressions are publicly available at[1], while examples of progressions contained in the dataset, as well of a few experiments, can be listened on the accompanying website[2]. When generating the audio excerpts corresponding to the chord progressions, an additional note was added in the lower register, always corresponding to the root of the chord. The voicings were implemented through the same model used in [34, 55] that combines voice-leading rules with a Viterbi algorithm.

We will first present an experiment aimed at examining the output chords generated by the trained models, then we present a listening test aimed at monitoring if the complexity values indicated by the latent space are consistent with the human perception.

[1] https://github.com/DavideGioiosa/cvae-chord-generation-complexity

[2] https://lucacoma.github.io/vae_complexity_website/

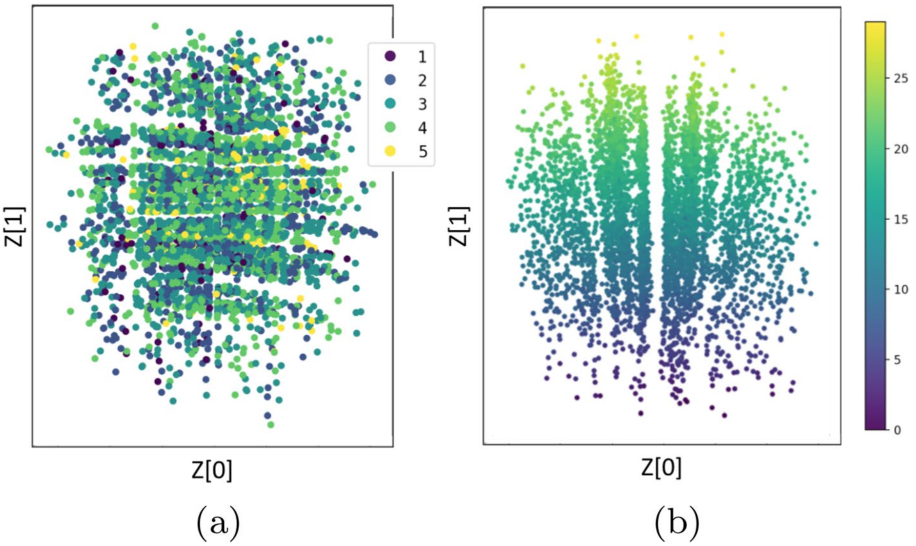


Fig. 7 Latent space representation obtained through the CVAE (**a**) and RVAE (**b**). The colors represent the different harmonic complexity bins *c* associated with the input progression, the values are from 1 to 30 in the case of RVAE (**b**) and from 1 to 5 for the CVAE (**a**)

6.1 Examples of generated chord progressions

We present an experiment assessing which type of chords are generated by the latent spaces obtained by training the CVAE and RVAE models. Specifically, we gave as input to both models the sequence corresponding to a chord progression and we varied the complexity level, by changing the corresponding class, for what concerns the CVAE, and by moving along the disentangled axis, for what concerns the RVAE. In order to make the comparison as easy as possible, in both cases we considered the aggregated complexity bins. It is important to notice that, while in this experiment we modify the complexity of the generated chords when keeping fixed the chord provided as input to the encoder, the model was not trained in order to enforce consistency between input and output chords when varying the complexity level. In fact, this generation procedure, is meant only as a preliminary experiment aimed at assessing the consistency of the generation capabilities of the model, while the general deployment of the model generates chords directly using the latent space as input to the CVAE and RVAE decoders. The input progression consisted of the following chords: Cmin Gmaj Gmin Fmin Cmin, corresponding to an aggregated complexity bin of 0. In Fig. 8, we depict the sheet music representation of the obtained progressions using CVAE (left column) and RVAE (right column) when varying the aggregated complexity bin from 1 to 5 (top to bottom row), it is possible to listen to the corresponding audio tracks on the accompanying website. We can see that a certain level of consistency with respect to the input chord is maintained using both the RVAE and CVAE, since in both cases the sequences start and end with the tonic Cmin, with the other possibility being Cmaj. When considering the lowest aggregated complexity level, as expected, the CVAE reconstructs exactly the input chord, since the model is trained using also the discrete complexity classes. With the RVAE model, instead, discrete classes correspond to intervals on the disentangled axis, therefore the generated chord progression is different from the input one. By inspecting the sequences, we can see that when increasing the complexity level, sequences become gradually different from the input progression adding also out-of-key chords, in accordance with what is expected from the considered harmonic complexity model (i.e., less expected chords result in higher harmonic complexity levels).

6.2 Listening test

In order to evaluate the effectiveness of the proposed model, we performed a listening test aimed at understanding if the perceived complexity of the generated chord progressions was coherent with the complexity selected during the generation part. The participants had to evaluate the complexity of the generated chord progression and indicate the class of complexity to which they thought it belong.

Before performing the actual listening test, the participants were profiled according to their musical background, through the self-report questionnaire of the Goldsmiths Musical Sophistication Index (GMSI) [62]. The test consists of 38 questions with seven-point scale answers for each question. The answers are then combined to form 5 sub-factors (active engagement, perceptual abilities, musical training, singing abilities, emotions)

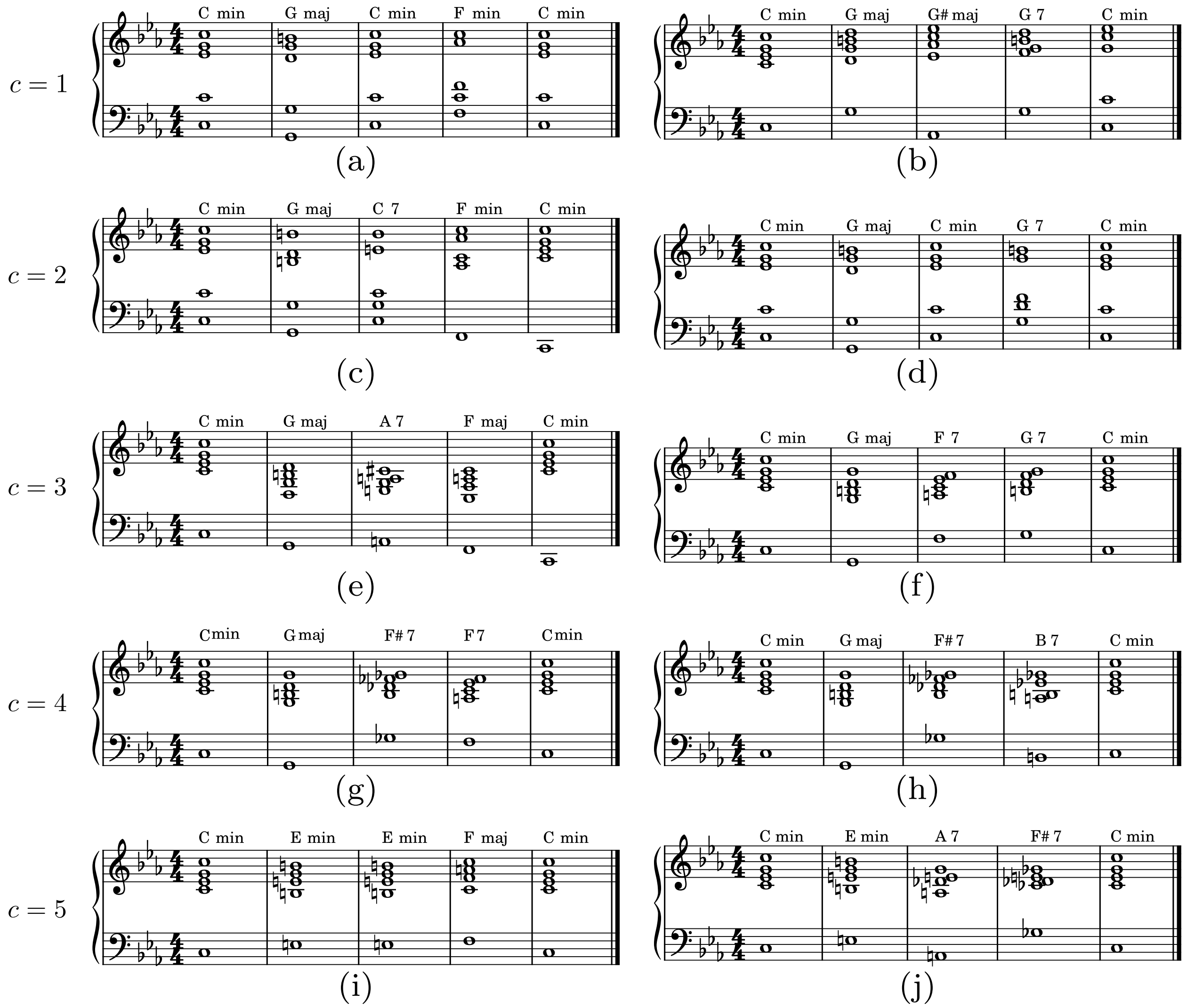


Fig. 8 Sequences generated by varying the aggregated complexity from 1 to 5 (top to bottom row, as indicated in the column on the left) with input chord sequence Cmin Gmaj Cmin Fmin Cmin, using CVAE (left column) and RVAE (right column)

and finally one general factor (general music sophistication factor). In Fig. 9 we show a histogram plot representing the GMSI of the participants.

Finally, we investigate a possible correlation between users' musical expertise expressed through the GMSI questionnaire and the ratings expressed. We used the Pearson correlation and the Spearman correlation to evaluate the musical level of a user in relation to the agreement values expressed in the perceptual test. We tested the correlation between the GMSI and the evaluation of the samples in relation to the different complexity classes and between the ratings given to the single sequences. The idea was to evaluate if users with similar values of musical knowledge express similar ratings on the audio samples. The results show that no significant correlation is present. In fact, in the ratings of the chord progressions, the users with close values of GMSI expressed discordant opinions (as in the example in Fig. 10). This seems to suggest that when evaluating the perceived complexity a subject's judgment does not depend only on his general musical knowledge, but also on other factors, such as familiarity with the type of music considered. We generated a total number of 80 test chord sequences, 40 for the CVAE model and 40 for the RVAE model. Each of these belonged to 5 possible complexity classes (from 1 to 5) evenly distributed. The sequences of chords used for the listening test are computed using the procedures described in Sections 5.2.2 and 5.3.2 for what concerns the CVAE and RVAE models, respectively. For simplicity, in the case of the RVAE,

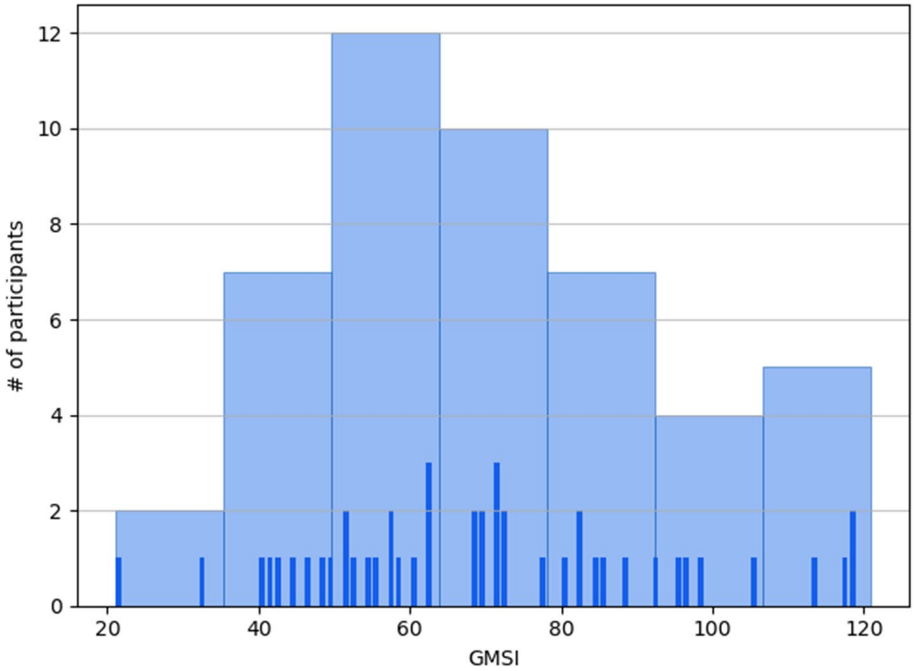


Fig. 9 The number of participants in the listening test grouped by musical expertise using the standard self-report questionnaire General Musical Sophistication Index (Gold MSI v1.0). High values of this measure indicate better musical skills and expertise of a user

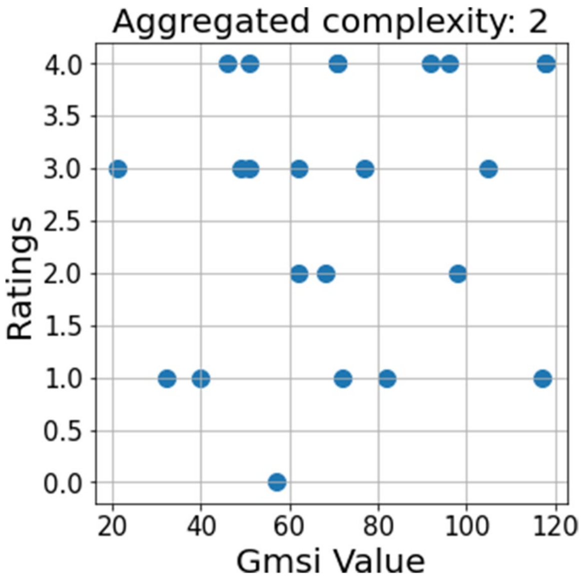


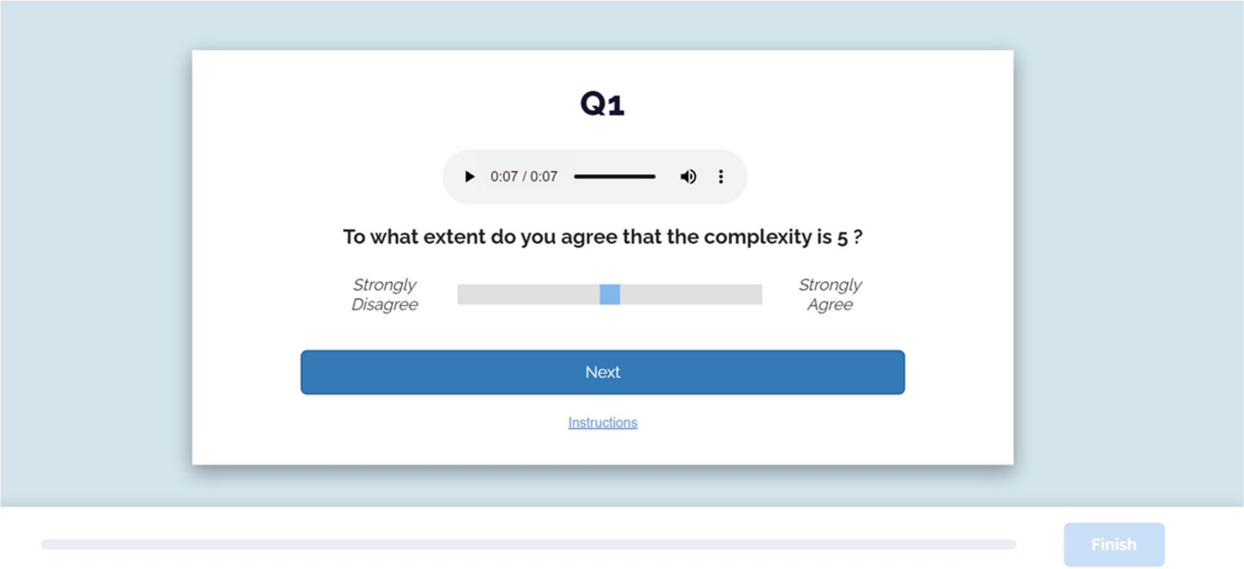
Fig. 10 Ratings of the users (*y*-axis) in relation to their GMSI value (*x*-axis) for an audio sample used in the test. No clear correlation has been highlighted between these two values

we used starting chords in order to find the coordinate corresponding to the disaligned complexity axis, and then moved along the other coordinate to generate other chords with similar complexity values. The listening test was organized as follows. In order to start acquainting themselves with the various complexity classes, needed in order to properly rate the progressions, the participants were able to hear 6 chord sequences extracted from the training set, before starting the actual experiments. Specifically, they listened to 3 progressions corresponding to the lowest complexity value and 3 corresponding to the highest one. In order to maintain a reference level, these progressions were listenable also during the rest of the experiment.

Then the users were able to listen to a series of chord progressions paired with a proposed complexity class and were asked to rate if the complexity value was right. This value is expressed using the Likert scale scores from 0 to 4, where 4 represents the value “strongly agree,” 3 “agree,” 2 “neither agree nor disagree,” 1 “disagree,” and 0 “strongly disagree.” In addition, if the user’s rating is less than 2, it is also asked to specify if the perceived complexity is greater or lower than the one indicated.

We considered a total of 80 sequences that are initially shuffled, then proposed to the participants using a Round Robin algorithm that sorts them by usage. The first two questions of the listening test are used to familiarize with the user interface and are not recorded.

Furthermore, the 20% of the proposed complexity values in the questionnaire are purposely wrong. This choice was made to reduce the Response Bias, in particular the acquiescence one [63], to avoid a user’s tendency to always express agreement answers. Obviously, we do not



To what extent do you agree that the complexity is 5 ?

Fig. 11 The listening test part of the experiment: the participants are asked to express their level of agreement to the indicated complexity value provided for each chord progression

use these ratings on incorrect values in the evaluation of the results.

The test was developed as a public web application, an example of the interface is shown in Fig. 11. We made several design choices to reduce the noise in the results. We first provided a precise description of the test structure and the questions that will have been asked to the user. The implementation of the GUI has been kept as simple and clean as possible. Furthermore, we proposed the test mainly to the people interested in our research, such as university students, researchers and professors, in order to reduce any risk of receiving random tests. Finally, we provided the possibility to send the results of the audio test after having done a minimum of 10 questions. This made the duration of the test variable (it lasts about 20 min by answering all the questions) depending on the listener's willingness to continue or send its ratings.

6.3 Listening test results

The test was taken by 47 participants and each chord sequence contained in the test dataset was evaluated approximately 30 times. As previously discussed, when a user expresses a value of disagreement with respect to the complexity estimated by our models, they have to specify whether they perceived the complexity to be higher or lower than the one indicated.

When analyzing the ratings provided by the participants on the chord sequences generated by the two models, we identified a series of chord progressions perceived as "ambiguous," meaning that there is not a clear consensus among the users that expressed disagreement (more than 50% of the participants who evaluated those audio samples) regarding the complexity indicated by our models. As an example, in Fig. 12, we show two sample ratings, corresponding to two chord progressions whose complexity value was perceived by the users as different from the one provided. The ratings shown in Fig. 12a are related to a chord progression whose proposed complexity value was perceived by approximately 93% as higher than the value proposed by our model. Due to the consensus on the rating of the progression, we define this rating as *non-ambiguous*. Instead, in the ratings related to another chord progression, shown in Fig. 12b, the participants do not seem to have a common opinion since around 46% of the users indicate that the complexity value should be minor, while 54% suggest it to be higher. We denote such rating as *ambiguous* since no clear consensus can be drawn. More specifically, we classify the rating of a chord progression as ambiguous when the difference between higher and lower values is less than 33.3%.

The goal of the two proposed models is to condition the generation of chord sequences according to an indicated value of harmonic complexity. Since the concept of complexity and harmony cannot be defined as universal among all cultures, together with the majority of music-related concepts [64], both the original ratings of complexity, presented in [31] and the evaluations performed in this paper are based on a socio-cultural group rooted in Western music culture.

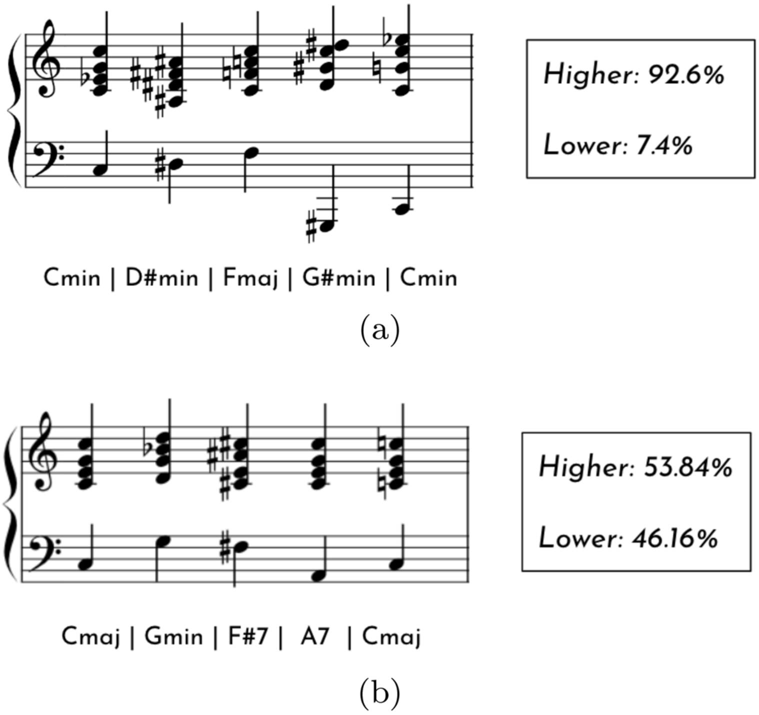

Higher: 92.6%
Lower: 7.4%
Cmin | D#min | Fmaj | G#min | Cmin
(a)
Higher: 53.84%
Lower: 46.16%
Cmaj | Gmin | F#7 | A7 | Cmaj
(b)

Fig. 12 Analysis of participants' negative ratings: among users who disagree with the complexity value indicated by our models, we evaluate how many of them consider that complexity has a different value from the one indicated in the test. In the first example (**a**), most users express that the complexity is higher. In the second (**b**), about half indicates a higher value, while the other half assigns a lower complexity. The case described in **b** is defined as "ambiguous" sample

For this reason, we identified and excluded from the analysis of the collected data 10 chord sequences generated by the RVAE and 3 by the CVAE, that present this ambiguity in the users' disagreement ratings.

In Fig. 13, we show the histogram related to the ratings expressed by the users for the 5 complexity classes over the chord progressions, excluding the ambiguous samples. The results show that more than 61% of the evaluations provided by the participants agreed with the complexity values used for generation through our models, of which 31.8% responded *agree* and 29.4% *strongly agree* (values "3" and "4," respectively, on the Likert scale). Approximately 9% responded *neither agree nor disagree*, while the remaining 24.6% chose *disagree* and only 5.2% *strongly disagree*. The percentage of disagreement has a significant value; however, this result was predictable given that complexity is a subjective parameter, as previously described. Despite this difficulty, the percentage of strong disagreement has a very low value and the results show that the two neural networks are capable of modeling complexity as a parameter for conditioning the chord generation process.

We proceed to analyze the same data by splitting the ratings on the audio samples according to the model from which they were generated. We describe these evaluations by dividing the sequences generated by the model of RVAE and CVAE in Table 2. The RVAE model shows slightly higher results than the CVAE one, obtaining about 64% (32.57% agree and 31.2% strongly agree) of positive evaluations compared to 59.3% (31.3% agree and 28% strongly agree) of the CVAE.

We then conduct a third analysis of the ratings expressed by the participants with respect to the complexity classes of the sequences generated by the two models. Results are reported in Table 3 for what concerns CVAE and in Table 4 for RVAE. The results show that the RVAE model performs better when generating chord sequences with low complexity (1 to 3). As an example, RVAE generates sequences producing about 67.5% agreement in the class of 3 versus 20.1% disagreement (of which only 3.3% strongly disagree). In contrast, the CVAE model performs better for high complexities (4 and 5). In particular, for what concerns the highest complexity level it obtains more than 78.6% agreement of the users.

7 Conclusion

In this paper, we have presented two techniques for conditional chord progression generation based on harmonic complexity, grounded in the Western culture

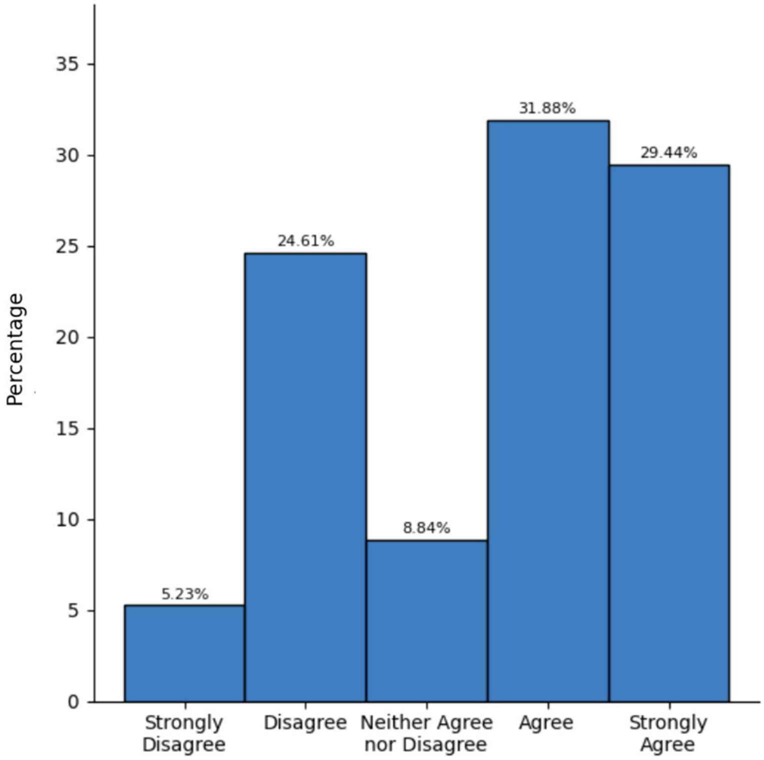
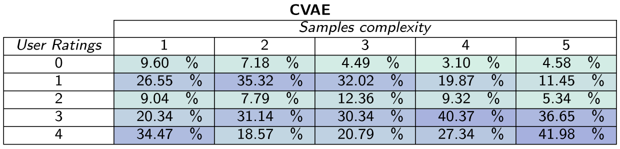


Fig. 13 Histogram of the 5 Likert scale values expressed by the participants to evaluate the level of agreement with the complexity value we indicated in the listening test

Table 2 Ratings of the participants on the progressions generated by the CVAE and RVAE models. The values are expressed using a 5-value Likert scale, from 0 to 4. The percentages indicate the average number of user ratings for a possible value out of the total, where 0 means Strongly Disagree and 4 is Strongly Agree

User ratings	CVAE	RVAE
0	5.89%	4.41%
1	25.79%	23.13%
2	8.96%	8.67%
3	31.32%	32.57%
4	28.00%	31.20%

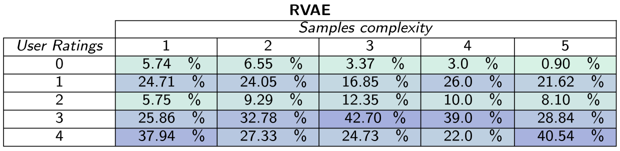
Table 3 Ratings of the participants on the audio samples generated by the CVAE w.r.t. their complexity values. The percentages indicate the average number of ratings for each value of the Likert scale out of the total for each of the complexity classes



CVAE

	Samples complexity				
User Ratings	1	2	3	4	5
0	9.60 %	7.18 %	4.49 %	3.10 %	4.58 %
1	26.55 %	35.32 %	32.02 %	19.87 %	11.45 %
2	9.04 %	7.79 %	12.36 %	9.32 %	5.34 %
3	20.34 %	31.14 %	30.34 %	40.37 %	36.65 %
4	34.47 %	18.57 %	20.79 %	27.34 %	41.98 %

Table 4 Ratings of the participants on the audio samples generated by the RVAE w.r.t. their complexity values. The percentages indicate the average number of ratings for each value of the Likert scale out of the total for each of the complexity classes



RVAE

	Samples complexity				
User Ratings	1	2	3	4	5
0	5.74 %	6.55 %	3.37 %	3.0 %	0.90 %
1	24.71 %	24.05 %	16.85 %	26.0 %	21.62 %
2	5.75 %	9.29 %	12.35 %	10.0 %	8.10 %
3	25.86 %	32.78 %	42.70 %	39.0 %	28.84 %
4	37.94 %	27.33 %	24.73 %	22.0 %	40.54 %

perception. We considered an already existing definition of complexity, based on the cognitive expectation in sequences of chords. More specifically, we proposed a CVAE and RVAE architecture able to condition a latent space based on the harmonic complexity values of the chord progressions. We perform a listening test through which we evaluate the correspondence between the complexity values considered in the generation process and the ones perceived by the participants to the test. Results show that a certain degree of accordance between the generated sequences and the perceived complexity is present. These findings motivate us to further developments, both with respect to the type of network architectures (Generative Adversarial Models, Flow-based Generative Models, etc.) and to the type of sequence considered, e.g., by extending the model to consider complete songs and different instruments.

Abbreviations

VAE	Variational Autoencoder
CVAE	Conditional Variational Autoencoder
RVAE	Regressor-based Variational Autoencoder
VQ-VAE	Vector-Quantized Variational Autoencoder
LSTM	Long Short-Term Memory
SGD	Stochastic Gradient Descent
GMSI	Goldsmiths Musical Sophistication Index

Acknowledgements

Not applicable.

Authors' contributions

LC: conceptualization, main writing and research oversee. DG: code implementation, conceptualization, writing. MZ: conceptualization, writing, research oversee. FA: research oversee and manuscript review. AS: research oversee and manuscript review. All authors read and agreed to the submitted version of the manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request. The code used to perform the experiments is fully available at https://github.com/DavideGioiosa/cvae-chord-generation-complexity.

Declarations

Ethics approval and consent to participate

The authors approve and consent to participate.

Consent for publication

The authors consent for publication.

Competing interests

The authors declare that they have no competing interests.

Received: 17 September 2022 Accepted: 21 April 2023
Published online: 15 May 2023

References

1. L. Hiller Jr, L. Isaacson, in *Audio Engineering Society Convention 9*. Musical composition with a high speed digital computer (Audio Engineering Society, New York, 1957)
2. D. Conklin, I.H. Witten, Multiple viewpoint systems for music prediction. J. New Music. Res. **24**(1), 51–73 (1995)
3. F. Pachet, P. Roy, Musical harmonization with constraints: A survey. Constraints **6**(1), 7–19 (2001)
4. A.R. François, I. Schankler, E. Chew, Mimi4x: An interactive audio-visual installation for high-level structural improvisation. Int. J. Arts Technol. **6**(2), 138–151 (2013)
5. J.P. Briot, F. Pachet, Deep learning for music generation: challenges and directions. Neural Comput. Applic. **32**(4), 981–993 (2020)
6. R.A. Fiebrink, B. Caramiaux, The Machine Learning Algorithm as Creative Musical Tool, in *The Oxford Handbook of Algorithmic Music*. (Oxford University Press, Oxford, 2018)
7. C. Plut, P. Pasquier, Generative music in video games: State of the art, challenges, and prospects. Entertain. Comput. **33**, 100337 (2020)
8. D. Temperley, *The cognition of basic musical structures* (MIT press, Cambridge, 2004)
9. M.M. Marin, A. Lampatz, M. Wandl, H. Leder, Berlyne revisited: Evidence for the multifaceted nature of hedonic tone in the appreciation of paintings and music. Front. Hum. Neurosci. **10**, 536 (2016)
10. A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio. (2016). arXiv preprint arXiv:1609.03499
11. S. Dieleman, A. van den Oord, K. Simonyan, The challenge of realistic music generation: modelling raw audio at scale. Adv. Neural Inf. Process. Syst. **31**, 7989–7999 (2018)
12. C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.Z.A. Huang, S. Dieleman, E. Elsen, J. Engel, D. Eck, Enabling factorized piano music modeling and generation with the MAESTRO dataset. (Paper presented at Proc. of the *International Conference on Learning Representations*, New Orleans, 2019)
13. D. Eck, J. Schmidhuber, in *Proceedings of the 12th IEEE workshop on neural networks for signal processing*. Finding temporal structure in music: Blues improvisation with lstm recurrent networks (IEEE, 2002), pp. 747–756
14. S. Oore, I. Simon, S. Dieleman, D. Eck, K. Simonyan, This time with feeling: Learning expressive musical performance. Neural Comput. Applic. **32**(4), 955–967 (2020)
15. G. Hadjeres, F. Pachet, F. Nielsen, in *International Conference on Machine Learning*. Deepbach: a steerable model for bach chorales generation (PMLR, 2017, International Machine Learning Society (IMLS), Princeton), pp. 1362–1371
16. C.Z.A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A.M. Dai, M.D. Hoffman, D. Eck, Music transformer: Generating music with long-term structure. (2018). arXiv preprint arXiv:1809.04281
17. A. Roberts, J. Engel, D. Eck, Hierarchical variational autoencoders for music. (Paper presented NIPS Workshop on Machine Learning for Creativity and Design, Long Beach, 2017)
18. D. Pressnitzer, C. Suied, S. Shamma, Auditory scene analysis: the sweet music of ambiguity. Front. Hum. Neurosci. (5), 158 (2011)
19. M. Buccoli, P. Bestagini, M. Zanoni, A. Sarti, S. Tubaro, in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. Unsupervised feature learning for bootleg detection using deep learning architectures (IEEE, 2014), pp. 131–136
20. M. Buccoli, M. Zanoni, F. Setragno, F. Antonacci, A. Sarti, in *2015 23rd European Signal Processing Conference (EUSIPCO)*. An unsupervised approach to the semantic description of the sound quality of violins (IEEE, 2015), pp. 2004–2008
21. M. Buccoli, M. Zanoni, A. Sarti, S. Tubaro, D. Andreoletti, in *2016 24th European Signal Processing Conference (EUSIPCO)*. Unsupervised feature learning for music structural analysis (IEEE, 2016), pp. 993–997
22. D. Herremans, E. Chew, Morpheus: generating structured music with constrained patterns and tension. IEEE Trans. Affect. Comput. **10**(4), 510–523 (2017)
23. O. Cífka, U. Şimşekli, G. Richard, Groove2groove: One-shot music style transfer with supervision from synthetic data. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 2638–2650 (2020)
24. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Adv. Neural Inf. Process. Syst. **30**, 6000–6010 (2017)
25. Y.J. Shih, S.L. Wu, F. Zalkow, M. Muller, Y.H. Yang, Theme transformer: Symbolic music generation with theme-conditioned transformer. IEEE Trans. Multimed. (2022)
26. A. Roberts, J. Engel, C. Raffel, C. Hawthorne, D. Eck, in *International conference on machine learning*. A hierarchical latent vector model for learning long-term structure in music (PMLR, International Machine Learning Society (IMLS), Princeton, 2018), pp. 4364–4373
27. A. Valenti, A. Carta, D. Bacciu, Learning style-aware symbolic music representations by adversarial autoencoders, in *ECAI 2020*. (IOS Press, Amsterdam, 2020), pp.1563–1570
28. R. Guo, I. Simpson, T. Magnusson, C. Kiefer, D. Herremans, A variational autoencoder for music generation controlled by tonal tension. (2020). arXiv preprint arXiv:2010.06230

29. J. Grekow, T. Dimitrova-Grekow, Monophonic music generation with a given emotion using conditional variational autoencoder. IEEE Access **9**, 129088–129101 (2021)
30. C. Weiß, M. Mauch, S. Dixon, M. Müller, Investigating style evolution of western classical music: A computational approach. Music. Sci. **23**(4), 486–507 (2019)
31. B. Di Giorgi, S. Dixon, M. Zanoni, A. Sarti, A data-driven model of tonal chord sequence complexity. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(11), 2237–2250 (2017)
32. K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models. Adv. Neural Inf. Process. Syst. **28**, 3483–3491 (2015)
33. Q. Zhao, E. Adeli, N. Honnorat, T. Leng, K.M. Pohl, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Variational autoencoder for regression: Application to brain aging analysis (Springer, New York, 2019), pp. 823–831
34. B. Di Giorgi, M. Zanoni, A. Sarti, S. Tubaro, in *nDS'13; Proceedings of the 8th International Workshop on Multidimensional Systems*. Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony (VDE, Berlin, 2013), pp. 1–6
35. S. Streich, P. Herrera, in *Audio Engineering Society Conference: 25th International Conference: Metadata for Audio*. Towards describing perceived complexity of songs: computational methods and implementation (Audio Engineering Society, New York, 2004)
36. J. Pressing, in *Proceedings of the 4th Conference of the Australasian Cognitive Science Society*. Cognitive complexity and the structure of musical patterns (1999)
37. D.E. Berlyne, Aesthetics and psychobiology. J. Aesthet. Art Crit. **31**(4) 553 (1973)
38. R.G. Heyduk, Rated preference for musical compositions as it relates to complexity and exposure frequency. Percept. Psychophys. **17**(1), 84–90 (1975)
39. S. Streich et al., *Music complexity: a multi-faceted description of audio content* (Universitat Pompeu Fabra, Barcelona, 2006)
40. T. Eerola, A.C. North, in *Proceedings of the Sixth International Conference on Music Perception and Cognition. Keele, Staffordshire, UK: Department of Psychology.*. Expectancy-based model of melodic complexity (2000)
41. T. Parmer, and Y.-Y. Ahn, Evolution of the Informational Complexity of Contemporary Western Music. In *Proceedings of the 20th annual conference of the International Society for Music Information Retrieval.* (ISMIR, Delft, 2019), pp. 175–182
42. M. Rohrmeier, T. Graepel, in *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval*. Comparing feature-based models of harmony (Springer, New York, 2012), pp. 357–370
43. S. Kostka, D. Payne, *Tonal harmony* (McGraw-Hill Higher Education, New York City, 2013)
44. S. Koelsch, T. Gunter, A.D. Friederici, E. Schröger, Brain indices of music processing: "nonmusicians" are musical. J. Cogn. Neurosci. **12**(3), 520–541 (2000)
45. W.P. Malm, *Music cultures of the Pacific, the Near East, and Asia*, vol. 2 (Prentice Hall Inc, New Jersey, 1977)
46. T. Fujishima, Real-time chord recognition of musical sound: A system using common lisp music. Proc. ICMC, Oct. 1999 pp. 464–467 (1999)
47. M. Mauch and M. Levy, "Structural change on multiple time scales as a correlate of musical complexity," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.* (2011) pp. 489–494.
48. L. Maršık, J. Pokorny, M. Ilcık, in *Procedings of the Annual International Workshop on Databases, Texts, Specifications, and Objects (DATESO)*. Towards a harmonic complexity of musical pieces (CEUR-WS, Aachen, 2014), pp. 1–12
49. L. Maršík, J. Pokornyy, M. Ilcík, in *Proceedings of the 14th conference Information Technologies-Applications and Theory*. Improving music classification using harmonic complexity (2014), pp. 13–17
50. F. Pachet, "Surprising harmonies," Int. J. Comput. Anticipatory Syst. **4**, 139–161 (1999)
51. R.P. Whorley, G.A. Wiggins, C. Rhodes, M.T. Pearce, Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem. J. New Music Res. **42**(3), 237–266 (2013)
52. J. Cleary, I. Witten, Data compression using adaptive coding and partial string matching. IEEE Trans. Commun. **32**(4), 396–402 (1984)
53. L. Rabiner, B. Juang, An introduction to hidden markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986). https://doi.org/10.1109/MASSP.1986.1165342
54. Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019)
55. F. Foscarin, Chord sequences: Evaluating the effect of complexity on preference. Master's thesis, Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB) (2017)
56. L. Maršík, Music harmony analysis: towards a harmonic complexity of musical pieces, Master's thesis. Department of Computer Science, Comenius University in Bratislava (2017)
57. D.P. Kingma, M. Welling, in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Auto-Encoding Variational Bayes (2014)
58. D.J. Rezende, S. Mohamed, D. Wierstra, in *International conference on machine learning*. Stochastic backpropagation and approximate inference in deep generative models (PMLR, nternational Machine Learning Society (IMLS), Princeton, 2014), pp. 1278–1286
59. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
60. I. Higgins, L. Matthey, A. Pal, C.P. Burgess, X. Glorot, M.M. Botvinick, S. Mohamed, A. Lerchner, in *International conference on learning representations (ICLR)*. beta-vae: Learning basic visual concepts with a constrained variational framework (2017)
61. Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives. IEEE Trans. Pattern. Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
62. D. Müllensiefen, B. Gingras, J. Musil, L. Stewart, The musicality of non-musicians: an index for assessing musical sophistication in the general population. PLoS ONE **9**(2), e89642 (2014)
63. P.M. Bentler, D.N. Jackson, S. Messick, Identification of content and style: a two-dimensional interpretation of acquiescence. Psychol. Bull. **76**(3), 186 (1971)
64. J.H. McDermott, A.F. Schultz, E.A. Undurraga, R.A. Godoy, Indifference to dissonance in native amazonians reveals cultural variation in music perception. Nature **535**(7613), 547–550 (2016)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.